

Claims

1. (currently amended) A computer-readable medium storing computer-executable instructions for causing a computer system programmed thereby to perform a method comprising:
 - detecting a reference to an interface, wherein a software program comprises plural units, and wherein one of the plural units exposes the interface;
 - determining if the interface has been wrapped by checking a data structure that tracks interface wrapping for the plural units of the software program by determining if there is an existing entry for the interface in the data structure;
 - if the interface has been wrapped, returning a reference to ~~an~~ the existing entry for the interface from the data structure, wherein the existing entry is a reference to an existing interface wrapper related to the interface which associates the interface with a unit identity for the unit that exposes the interface;
 - if the interface has not been wrapped,
 - discovering the unit identity of the unit that exposes the interface;
 - adding a new entry to the data structure, wherein the new entry is a reference to a new interface wrapper related to the interface which associates the interface with the discovered unit identity; and
 - returning a reference to the new entry.
2. (canceled)
3. (previously presented) The computer-readable medium of claim 1 wherein the step of detecting comprises:
 - noting one or more return parameters from a called function; and
 - parsing the one or more return parameters to detect the reference to the interface.
4. (previously presented) The computer-readable medium of claim 3 wherein the called function is a unit creation function.

5. (previously presented) The computer-readable medium of claim 1 wherein the step of detecting comprises:

noting one or more outgoing parameters to a called function; and
parsing the one or more outgoing parameters to detect the reference to the interface.

6. (previously presented) The computer-readable medium of claim 1 wherein the data structure comprises a hash table, and wherein the step of determining comprises:

hashing the detected reference.

7. (currently amended) The computer-readable medium of claim 1 wherein the data structure is comprises a hash table, and wherein the step of adding the new entry comprises:

creating the new entry in the hash table, ~~wherein the new entry associates the interface with the discovered unit identity.~~

8. (currently amended) The computer-readable medium of claim 1 wherein the data structure comprises a hash table for associating interfaces with interface wrappers, wherein the existing entry is for ~~an~~ the existing interface wrapper for the interface in the hash table, and wherein the step of determining comprises:

hashing the detected reference.

9. (canceled)

10. (currently amended) The computer-readable medium of claim 1 wherein the data structure comprises a hash table for associating interfaces with interface wrappers, ~~and wherein the new entry is for a new interface wrapper that stores the unit identity.~~

11. (previously presented) The computer-readable medium of claim 1 wherein a local variable stores data comprising the unit identity of the unit that exposes the interface, and wherein the step of discovering the unit identity comprises noting the value stored in the local variable.

12. (previously presented) The computer-readable medium of claim 11 wherein an instrumentation system provides the unit identity of the unit that exposes the interface.

13. (previously presented) The computer-readable medium of claim 1 wherein the method further comprises:

if the interface has been wrapped, verifying the unit identity of the unit that exposes the interface.

14. (previously presented) The computer-readable medium of claim 1 wherein the method further comprises:

detecting a communication passing through the interface;

measuring the size of the communication;

determining the unit identity of the unit that exposes the interface from the returned reference to the existing or new entry of the data structure; and

associating the measured size with the unit that exposes the interface.

15. (previously presented) The computer-readable medium of claim 1 wherein the method further comprises:

receiving from a client unit a call to a unit activation function for a unit to be activated among the plural units;

determining a unit identity of the client unit from the data structure;

classifying the unit to be activated based upon the unit identity of the client unit, resulting in a classifier;

determining a location in a distributed computing environment using the classifier; and

routing the call to the location.

16. (currently amended) A computer-readable medium storing computer-executable instructions for causing a computer system programmed thereby to perform a method comprising:

on receiving a reference to an interface as a return parameter from a function call,

determining if the interface is wrapped by checking a data structure that tracks interface

wrapping for plural components of software for an entry related to the interface which associates the interface with one of the plural components that, ~~wherein one of the plural components~~ exposes the interface;

if the interface is not wrapped,
discovering component identity of the component that exposes the interface; and
adding a new entry to the data structure, wherein the new entry associates the interface with the discovered component identity.

17. (canceled)

18. (previously presented) The computer-readable medium of claim 16 wherein the data structure comprises a hash table, and wherein the step of determining comprises:
hashing the received reference.

19. (currently amended) The computer-readable medium of claim 16 wherein the data structure comprises a hash table, wherein the step of adding the new entry comprises:
creating the new entry in the hash table, ~~wherein the new entry associates the interface with the discovered component identity.~~

20. (currently amended) A computer-readable medium storing computer-executable instructions for causing a computer system programmed thereby to perform a method comprising:
detecting a reference to an interface, wherein one of plural components of software exposes the interface;

determining if the interface is wrapped using a hash table, wherein if the interface is wrapped, a reference to an existing interface wrapper related to the interface is stored in the hash table, and wherein the existing interface wrapper stores a reference to instrumentation, the reference to the interface, and component identity of the component that exposes the interface;

if the interface is wrapped, providing to a client component a the reference to ~~an~~ the existing interface wrapper, ~~wherein the existing interface wrapper stores a reference to instrumentation, the reference to the interface, and component identity of the component that exposes the interface;~~

if the interface is not wrapped,
 creating a new interface wrapper for the interface, wherein the new interface wrapper stores the reference to the instrumentation and the reference to the interface;
 creating a new entry in the hash table, wherein the new entry ~~associates~~ relates the interface with the created new interface wrapper;
 discovering the component identity of the component that exposes the interface;
 storing in the new interface wrapper the component identity of the component that exposes the interface; and
 providing to the client component a reference to the new interface wrapper; and
receiving from the client component an invocation of the instrumentation through the provided reference to the existing or new interface wrapper.

21. (canceled)

22. (previously presented) The computer-readable medium of claim 20 wherein the step of determining if the interface is wrapped comprises:

 hashing the detected reference;

 if the detected reference hashes to the existing interface wrapper, returning the reference to the existing interface wrapper; and

 otherwise, returning a value that indicates the interface is not wrapped.

23. (previously presented) The computer-readable medium of claim 20 wherein the reference to instrumentation comprises a pointer to a table comprising at least one pointer to one or more instrumentation functions.

24. (previously presented) The computer-readable medium of claim 20 wherein the step of creating the new interface wrapper further comprises storing in the new interface wrapper a type description of the interface.

25. (previously presented) The computer-readable medium of claim 20 wherein a local variable stores the component identity of the component that exposes the interface, and wherein the step of discovering the component identity of the component that exposes the interface comprises noting the value stored in the local variable.

26. (previously presented) The computer-readable medium of claim 20 wherein the invocation comprises a communication from the client component directed towards the component that exposes the interface, the method further comprising:

measuring the size of the communication using the instrumentation;

associating the measured size with the component that exposes the interface and the client component; and

calling the component that exposes the interface.

27. – 66. (canceled)

67. (previously presented) The computer-readable medium of claim 1 wherein the computer-executable instructions are for automatic distributed partitioning system software.

68. (previously presented) The computer-readable medium of claim 1 wherein the reference to the interface is a pointer to the interface.

69. (previously presented) The computer-readable medium of claim 16 wherein the computer-executable instructions are for automatic distributed partitioning system software.

70. (previously presented) The computer-readable medium of claim 16 wherein the reference to the interface is a pointer to the interface.

71. (previously presented) The computer-readable medium of claim 20 wherein the computer-executable instructions are for automatic distributed partitioning system software.

72. (previously presented) The computer-readable medium of claim 20 wherein the reference to the interface is a pointer to the interface.

73. (canceled)